

MRC – THE THEORY OF LAYER-BASED DOCUMENT IMAGE COMPRESSION

Costin-Anton Boiangiu ^{1*}
Luiza Grigoraş ²

ABSTRACT

The concept of Mixed Raster Content describes a compound document image as being composed of several layers, each containing a part of its visual information. Usually, three layers are sufficient for classifying the types of content present in such an image: a foreground layer, a background layer, and a mask layer. In this context, MRC-based compression schemes promise to be more efficient than classical ones (where a single algorithm is used to compress the entire image), due to their implicit content-adaptive nature, because each layer can be compressed separately with a suitable algorithm (JPEG, JBIG etc.).

KEYWORDS: *MRC, Document Compression, Image Compression, Data Compression, Image Processing, OCR, Resampling Filters.*

INTRODUCTION

From the image processing field of research, image compression is of high interest nowadays, as performance (of transmission of images over the Internet or by fax) and storage issues (for online libraries, online databases of images) have become more prominent, with the increased rates of information exchange across electronic media.

There is a variety of compression algorithms and image formats, each being designed for a particular purpose and image type in mind (De Queiroz et al., 1999). For example, JPEG and JPEG2000 are designed for natural image compression (Rabbani and Joshi, 2002), while JBIG2 favors images with recurring symbols (Haneda and Bouman, 2011), such as document images.

A compression algorithm good for all types of images does not exist (De Queiroz et al., 1999; De Queiroz, 2005), but the variety of compression algorithms can be exploited by using a generalized framework, one which could adapt the algorithm to the characteristics of the image, at least to some degree. De Queiroz et al. (1999) stipulates that this could be accomplished by a compression algorithm based on the concept of Mixed Raster Content. This concept describes an image (comprising information in various forms: text, pictures, line art) as a composite of layers, each with different semantics and different visual and signal characteristics, correspondingly.

^{1*} corresponding author, Professor PhD Eng., Politehnica University of Bucharest, 060042 Bucharest, Romania, costin.boiangiu@cs.pub.ro

² Engineer, Politehnica University of Bucharest, 060042 Bucharest, Romania, luiza.grigoras@cti.pub.ro

MRC COMPRESSION BASICS

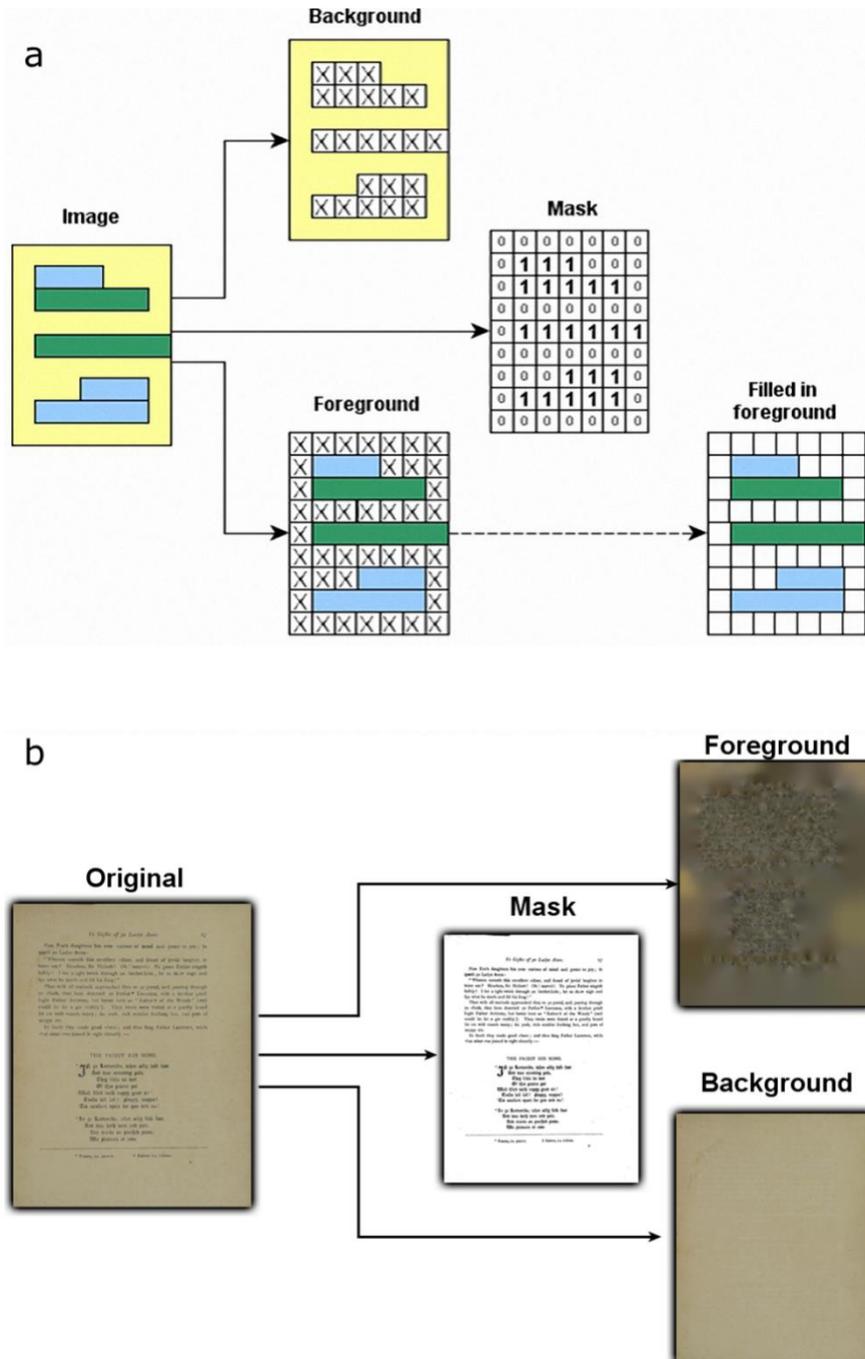


Figure 1. Simple MRC decomposition scheme: (a) Abstraction. After the decomposition process, many gaps are left in the foreground and background layers (marked with "X"). The bottom images show the data-filling of the foreground layer with a constant color (in this case, white). (b) Exemplification on a concrete image (layers obtained with proposed codec).

An MRC image can be decomposed into as many layers as one considers it necessary, but usually three layers are considered to be sufficient in categorizing image information and compressing it accordingly (ITU-T Recommendation T.44, 2005), as illustrated in Fig 1: the foreground layer (text color), the mask layer (identifying text and contours) and the background layer (background colors and embedded images). This decomposition of an image is most advantageous, allowing the separate compression of layers with the most suited algorithm for each one and at different compression ratios (Pavlidis, 2017). For example, the background layer may contain continuous-tone images and/or areas of constant color, which can be well compressed using JPEG or JPEG2000. The foreground layer usually contains only areas of constant color, and thus can be heavily compressed (Haneda and Bouman, 2011) using the same algorithms. However, the mask layer is a bi-level image, containing many characters, i.e., recurring symbols, for which JBIG or JBIG2 obtain better performances, as mentioned above. Therefore, MRC is implicitly adaptive and versatile (Mukherjee et al., 2002) and can unify several compression algorithms. Furthermore, it can be reduced to each one of them, when a single layer is used for the image (ITU-T Recommendation T.44, 2005).

Because of this decomposition, a pixel in the three-layered image has now 49 corresponding bits in MRC representation (if the image is represented in a 24 bits per pixel format) (De Queiroz et al., 1999; Mukherjee et al., 2002), but this increased size of the image is compensated by the final compression ratio, as a result of compressing each layer with a different algorithm. Thus, this method can compete with a single compression algorithm applied on the image as a whole and can even outperform it. The performance gain of a three-layered based MRC approach is obvious in the case of simple text documents, as outlined by De Queiroz (2005), while for continuous-tone images this approach is less suitable.

Several MRC-based codecs have been proposed: (De Queiroz, 2005) analyzed a JPEG-MMR-JPEG MRC encoder and proposed one based on JPEG2000; (Zaghetto and De Queiroz, 2007) propose another coding scheme based on the H.264 video compression standard that works very well for still images too and compared an MRC JPEG2000-based solution to it. In both solutions, JBIG2 was used for mask compression. In Mukherjee et al. (2002) an MRC-compliant codec using JPEG2000 and JBIG compression is analyzed; the emphasis is on the segmentation steps, which are specially designed to suit and encourage JPEG2000 compression.

There are also proprietary solutions based on image layer decomposition. The most famous example is DjVu, which uses a JBIG-like algorithm to encode the mask (Bottou et al., 1998) and a decomposition algorithm that is similar to that implied by MRC, except the fact that the foreground layer encompasses the mask layer, by preserving text (letter shapes) associated with its colors (De Queiroz, 2005).

Following (De Queiroz, 2005; Mukherjee et al., 2002; Zaghetto and De Queiroz, 2008), a general MRC compression scheme can be derived that would include several steps, among which the most important are: image decomposition into layers, data-filling of foreground and background layers and actual compression of each layer using a suitable compression algorithm. The decomposition step is considered to be the most important one (Haneda and Bouman, 2011). The pixels of the image are separated into two layers,

background and foreground, based on their corresponding color levels. The result of this stage is the mask layer, which acts like a sieve, making it possible to distinguish between the pixels belonging to each layer. A clear separation between the foreground and the background layer should be accomplished, in order for the subsequently applied compression algorithms to perform well (Haneda and Bouman, 2011). The data filling step is also important, because it prepares the layers for the actual compression, therefore greatly influencing compression performance. The empty regions resulted in the foreground and the background layers are filled in with a certain color (or colors), in order to make the layers compress well, according to a specific compression algorithm. To improve compression furthermore, the background and the foreground layers should not contain abrupt color transitions; the data-filling process must ensure the smoothness of the resulting image (De Queiroz, 2005; Mukherjee et al., 2002).

Many other refinements can be added to this scheme, in order to reduce layer size and therefore improve compression ratio (layer downsampling, region/stripe decomposition etc.). Due to the fact that they usually contain very smooth color transitions and few details, the foreground and the background layers can also be subsampled in order to reduce the size of the layer to be compressed (De Queiroz et al., 1999), without the risk of losing important information in the process. Better compression rates can be obtained by also preparing layers for the type of compression they are supposed to be put through. For example, for JPEG compression, an approach based on the decomposition of the image into blocks is advantageous (De Queiroz, 2005). These blocks are individually segmented, filled in and smoothed. For JPEG2000 compression, which is not block-based, "smoothness guarantees compactness" (Mukherjee et al., 2002). That is why the data-filling stage has to be treated carefully, as it plays an important part in obtaining a good compression ratio.

DATA-FILLING USING INTERPOLATION AND RESAMPLING

De Queiroz (2000) emphasized that the problem with a data-filling algorithm is to produce smooth transitions between non-empty pixels and empty pixels filled in with a custom color. Although this can be accomplished by various means, a simple method based on segmented filtering and an iterative method based on DCT was presented, whereas other papers (Lakhani and Subedi, 2006; De Queiroz, 2006) discuss methods based on wavelets. (Mukherjee et al., 2002) emphasized the importance of the data filling step in preparing layers for JPEG2000 compression and presented a data-filling algorithm which implies interpolating image pixels; smoothing an image at a high degree is obtained using a Gaussian filter.

On these grounds, we can relate to the field of image interpolation and resampling, in order to perform the data-filling of a layer. The interpolation and resampling steps can be united in a single one and be ideally performed by using a low-pass filter, which eliminates high-frequency components that may cause aliasing in the resulting image. The sinc function is the ideal low-pass filter (Smith, 1997): its finite impulse response has a perfectly rectangular shape; it has a transition band of width 0 (thus the steepest roll-off) and no ripples, neither in the passband nor in the stopband. The function has the form:

$$\text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (1)$$

Although any function can be used as a base function for interpolation (Thévenaz et al., 2000), all efforts were concentrated towards the sinc function, either to find good and easy-to-compute polynomial approximations for it or to find a practical form for using it. This function has infinite support, thus it cannot be used in practice as it is. Its domain has to be restricted. Directly truncating the function results in discontinuities at the boundaries of its restricted domain which causes severe artifacts in the filtered image; these include ringing, blurring, and aliasing (Mitchell and Netravali, 1988; Hauser et al. 2000). Another form of restricting it is by multiplying the function with a finite-support window function (apodization function) as stated by Thévenaz et al., (2000). The usage of a window over the sinc function guarantees smooth transitions to zero and thus eliminates discontinuities, reducing the sinc filter to a practical width (Smith, 1997).

In order to make a thorough comparison between the filters applied in MRC compression, we selected several from three main families: polynomial, exponential and windowed-sinc.

POLYNOMIAL AND EXPONENTIAL FILTERS

Filters in this generic family have the great advantage of being easy to compute. They also perform well (Thévenaz et al., 2000), trying to approximate the ideal sinc function and reducing ringing and aliasing, but at the cost of blurring. Higher order polynomials usually give better results (Lehmann et al., 1999). Table 1 lists the polynomial filters used for this study, with their corresponding mathematical form.

Table 1. Polynomial and exponential resampling filters.

Filter name	Filter function
Box	$f(x) = \begin{cases} 1, & x \leq 0.5 \\ 0, & x > 0.5 \end{cases}$
Triangle	$f(x) = \begin{cases} 1 - x , & x \leq 1 \\ 0, & x > 1 \end{cases}$
Hermite	$f(x) = \begin{cases} 2 x ^3 - 3x^2 + 1, & x \leq 1 \\ 0, & x > 1 \end{cases}$

Filter name	Filter function		
<p>Spline Cubic_H4_1 ($a = -1$) Cubic_H4_2 ($a = -\frac{1}{2}$) Cubic_H4_3 ($a = -\frac{3}{4}$) Cubic_H4_4 ($a = -\frac{2}{3}$)</p>	$f(x) = \begin{cases} (a+2) x ^3 - (a+3)x^2 + 1, & x \leq 1 \\ a x ^3 - 5ax^2 + 8a x - 4a, & 1 < x \leq 2 \\ 0, & x > 2 \end{cases}$		
<p>3rd order B-Spline (Quadratic)</p>	$f(x) = \begin{cases} -x^2 + 0.75, & x \leq 0.5 \\ 0.5(x - 1.5)^2, & 0.5 < x \leq 1.5 \\ 0, & x > 1.5 \end{cases}$		
<p>4th order B-Spline (Cubic B-Spline)</p>	$f(x) = \begin{cases} \frac{ x ^3}{2} - x^2 + \frac{2}{3}, & x \leq 1 \\ \frac{1}{6}(2 - x)^3, & 1 < x \leq 2 \\ 0, & x > 2 \end{cases}$		
<p>BC-family</p>	$f(x) = \begin{cases} \frac{1}{6}[(12 - 9B - 6C) x ^3 + (-18 + 12B + 6C)x^2 + 6 - 2B], & x \leq 1 \\ \frac{1}{6}[(-B - 6C) x ^3 + (6B + 30C)x^2] + \frac{1}{6}[(-12B - 48C) x + 8B + 24C], & 1 < x \leq 2 \\ 0, & x > 2 \end{cases}$		
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">B</td> <td style="width: 50%; text-align: center; padding: 5px;">C</td> </tr> </table>	B	C
B	C		
<p>Mitchell</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">$\frac{1}{3}$</td> <td style="width: 50%; text-align: center; padding: 5px;">$\frac{1}{3}$</td> </tr> </table>	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{1}{3}$		
<p>Notch</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">1.5</td> <td style="width: 50%; text-align: center; padding: 5px;">-0.25</td> </tr> </table>	1.5	-0.25
1.5	-0.25		
<p>Catmull-Rom</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">0</td> <td style="width: 50%; text-align: center; padding: 5px;">0.5</td> </tr> </table>	0	0.5
0	0.5		
<p>Robidoux</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">0.3782</td> <td style="width: 50%; text-align: center; padding: 5px;">0.3109</td> </tr> </table>	0.3782	0.3109
0.3782	0.3109		
<p>Robidoux Sharp</p>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center; padding: 5px;">0.2620</td> <td style="width: 50%; text-align: center; padding: 5px;">0.3690</td> </tr> </table>	0.2620	0.3690
0.2620	0.3690		

Filter name	Filter function
Gaussian	$f(x) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}, & x \leq W \\ 0, & x > W \end{cases}$

The box filter is the simplest interpolating filter and has a sinc-shaped finite impulse response, which makes it a poor low-pass filter (Parker et al., 1983). Results obtained with the box filter may be satisfactory in some cases, especially when downsampling (Thyssen, 2017). The triangle filter, also known as the bilinear interpolation filter, gives a smooth, natural gradient pass between pixels when upsampling, in contrast to the box filter (Thyssen, 2017). The Hermite interpolation function is actually the basic cubic two-point interpolation function, the simplest case of a cubic polynomial with boundary conditions settled (C0 and C1 continuity) as stated by Lehmann et al. (1999). Splines are "piecewise polynomials with pieces that are smoothly connected together" (Unser, 1999). They are easier to compute than sinc-based filter functions and are appropriate for multi-resolution approaches which imply construction of image pyramids. Values for constant a that have been established based on optimality principles include -1, -3/4, -2/3 and -1/2 (Lehman et al., 1999; Parker et al., 1983). B-Splines are often used in practice (Thévenaz et al., 2000). They are obtained by multiplying a base function (the rectangle function) with itself, several times, obtaining higher order B-Splines. B-Spline approximating filters perform the most blurring of an image (Thyssen, 2017).

BC-family filters have been deduced and discussed by Mitchell and Netravali (1988). Based on experimental results, they were able to identify filters for which the best compromise may be obtained, for some or even all types of image artifacts. The "Mitchell" (B = C = 1/3, ideal filter) and "Notch" filters have been suggested by Mitchell and Netravali (1988), Catmull-Rom has been discussed in the same paper and included in this family, while the last two filters have been recommended in Thyssen (2017).

From the exponential filters, we have selected the Gaussian filter, which has been used as a blurring filter in image processing for a long time now, being useful in Gaussian noise removal. The Gaussian filter function slowly descends toward zero, but generally, it can be considered zero for $x \geq W \geq 3\sigma$, where W is the chosen width.

WINDOWED-SINC FILTERS

In his article, Harris (1978), performs a thorough review and analysis of many window functions, accompanied by various suggestive plots in time and frequency domains. The following versions of window functions (Table 2) reflect their exact code implementation and follow mostly Harris's (1978) analysis, thus being defined on $[-N/2, N/2]$, where N is the filter width. We also use the notation $W = N/2$.

Table 2. Window functions.

Window function name	Filter function				
Lanczos Lanczos3 ($W = 3$) Lanczos4 ($W = 4$)	$f(x) = \begin{cases} \text{sinc}\left(\frac{x}{W}\right), & x \leq W \\ 0, & x > W \end{cases}$				
Cosine Cosine ($\alpha = 1$) Cosine3 ($\alpha = 3$)	$f(x) = \begin{cases} \cos^\alpha\left(\frac{\pi x}{2W}\right), & x \leq W \\ 0, & x > W \end{cases}$				
Generalized Cosine	$f(x) = \begin{cases} \sum_{i=0}^n a_i \cos\left(\pi i \frac{x}{W}\right), & x \leq W \\ 0, & x > W \end{cases}$				
	a1	a2	a3	a4	a5
	Hann	0.5	0.5	-	-
	Hamming	0.54	0.46	-	-
	Blackman	0.426590	0.496560	0.076849	-
	Nuttall	0.355768	0.487396	0.144232	0.012604
	Blackman-Nuttall	0.3635819	0.4891775	0.1365995	0.0106411
Blackman-Harris	0.35875	0.48829	0.14128	0.01168	-
Kaiser-Bessel	0.40243	0.49804	0.09831	0.00122	-
Flat Top	0.21557894	0.41663158	0.27726315	0.08357894	0.00694736
Welch	$f(x) = \begin{cases} 1 - \left(\frac{x}{W}\right)^2, & x \leq W \\ 0, & x > W \end{cases}$				
Parzen	$f(x) = \begin{cases} 6\left \frac{x}{W}\right ^3 - 6\left(\frac{x}{W}\right)^2 + 1, & x \leq \frac{W}{2} \\ 2\left(1 - \left \frac{x}{W}\right \right)^3, & \frac{W}{2} < x \leq W \\ 0, & x > W \end{cases}$				

Window function name	Filter function
Bohman	$f(x) = \begin{cases} \left(1 - \frac{ x }{W}\right) \cos\left(\frac{\pi x}{W}\right) + \frac{1}{\pi} \sin\left(\frac{\pi x}{W}\right), & x \leq W \\ 0, & x > W \end{cases}$
Gaussian (GaussianW)	$f(x) = \begin{cases} -x^2 & x \leq W \\ e^{2\sigma^2}, & \\ 0, & x > W \end{cases}$

The Lanczos window is built on a simple idea: a truncated version of the sinc function is used to window the sinc function. Lanczos3 is quite popular (Thyssen, 2017) and gives good qualitative results (Turkowski, 1990). The cosine window is the first member ($\alpha = 1$) in the family of functions presented in Table 2. More windows can be obtained by varying α ; the greater the α power, the smoother the window and the better the results, but with an increased width of the main lobe (Harris, 1978). Cosine, Hann, Hamming, Blackman, Welch, Parzen, Bohman and Gaussian (having a slightly different form than that of the Gaussian filter) window functions are taken from Harris's (1978) paper on windows. All the rest of the generalized cosine window coefficients are taken from WOLFRAM (2017). The Blackman-Harris window as referred here is actually the 4-termed Blackman-Harris window and Kaiser-Bessel is actually the sampled version of the Kaiser-Bessel window. These two windows have been recommended by Harris in the same paper.

The rest of this paper comprises three sections: the proposed compression scheme, with emphasis on the data-filling stage; the results of the codec evaluation on several document images, analyzing the effects and performances of the selected resampling filters applied in the data-filling stage; recommendations with regard to the filters and parameters which are best to be used in this compression scheme.

REFERENCES

- [1] Bottou, L., Haffner, P., Howard, P.G., Simard, P., Bengio, Y., LeCun, Y. (1998). *High-Quality Document Image Compression with DjVu*. J. Electron. Imaging, 7, pp. 410-425.
- [2] De Queiroz, R.L., Buckeley, R., Xu, M. (1999). *Mixed Raster Content (MRC) Model for Compound Image Compression*. In Proceedings of SPIE Visual Communications and Image Processing, volume (3653), pp. 1106-1117.
- [3] De Queiroz, R.L. (2000). *On Data Filling Algorithms for MRC Layers*. In Proceedings of the IEEE International Conference on Image Processing, volume (2), pp. 586-589, Vancouver, Canada.
- [4] De Queiroz, R.L. (2005). *Compressing Compound Documents*. In Barni, M. (ed.), *The Document and Image Compression Handbook*. Marcel-Dekker.

- [5] De Queiroz, R.L. (2006). *Pre-Processing for MRC Layers of Scanned Images*. In Proceedings of the 13th IEEE International Conference on Image Processing, pp. 3093-3096, Atlanta, GA, U.S.A.
- [6] Glasner, D., Bagon, S., Irani, M. (2009). *Super-Resolution from a Single Image*. In Proceedings of the 12th IEEE International Conference on Computer Vision, pp. 349-356, Kyoto, Japan.
- [7] Paşca L. (2013), *Hybrid Compression Using Mixed Raster Content, Smart Resampling Filters and Super Resolution*, Diploma Thesis, unpublished work (original author name Paşca L., actual name Grigoraş L.).
- [8] Haneda, E., Bouman, C.A. (2011). *Text Segmentation for MRC Document Compression*. IEEE Trans. Image Process, pp. 1611-1626.
- [9] Harris, F.J. (1978). *On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform*. Proc. IEEE, 66, pp. 55-83.
- [10] Hauser, H., Groller, E., Theussl, T. (2000). *Mastering Windows: Improving Reconstruction*. In Proceedings of the IEEE Symposium on Volume Visualization, pp. 101-108. Salt Lake City, UT, U.S.A.
- [11] *ITU-T Recommendation T.44 Mixed Raster Content (MRC)*, (2005).
- [12] Lakhani, G., Subedi, R. (2006). *Optimal Filling of FG/BG Layers of Compound Document Images*. In Proceedings of the 13th IEEE International Conference on Image Processing, pp. 2273-2276, Atlanta, GA, U.S.A.
- [13] Lehmann, T.M., Gönner, C., Spitzer, K. (1999). *Survey: Interpolation Methods in Medical Image Processing*. IEEE Trans. Med. Imaging, 18, pp. 1049-1075.
- [14] Minaee, S., Abdolrashidi, A., Wang, Y. (2015). *Screen content image segmentation using sparse-smooth decomposition*. 49th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, pp. 1202-1206. DOI: 10.1109/ACSSC.2015.7421331.
- [15] Minaee, S., Wang, Y. (2015). *Screen content image segmentation using least absolute deviation fitting*. 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, pp. 3295-3299. DOI: 10.1109/ICIP.2015.7351413.
- [16] Mitchell, D.P., Netravali, A.N. (1988). *Reconstruction Filters in Computer Graphics*. ACM SIGGRAPH Comput. Graph, 22, pp. 221-228.
- [17] Mtimet, J., Amiri, H. (2013). *A layer-based segmentation method for compound images*. 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), Hammamet, pp. 1-5. DOI: 10.1109/SSD.2013.6564005.
- [18] Mukherjee, D., Chrysafis, C., Said, A. (2002). *JPEG2000-Matched MRC Compression of Compound Documents*. Proceedings of the IEEE International Conference on Image Processing, volume (3), pp. 73-76.

- [19] Parker, J.A., Kenyon, R.V., Troxel, D.E. (1983). *Comparison of Interpolating Methods for Image Resampling*. IEEE Trans. Med. Imaging, 2, pp. 31-39.
- [20] Pavlidis, G. (2017). *Mixed Raster Content, Segmentation, Compression, Transmission*. Signals and Communication Technology Series, Springer Singapore, DOI: 10.1007/978-981-10-2830-4.
- [21] Rabbani, M., Joshi, R. (2002). *An overview of the JPEG 2000 still image compression standard*. Signal Process. Image Commun, 17, pp. 3-48.
- [22] Smith, S.W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*, 1st ed., pp. 285-296. California Technical Publishing, San Diego, CA, U.S.A.
- [23] Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. In Proceedings of the 9th International Conference on Doc Analysis and Recognition, volume (2), pp. 629-633, Curitiba, Parana, Brasil.
- [24] Thévenaz, P., Blu, T., Unser, M. (2000). *Image Interpolation and Resampling*. In *Handbook of Medical Imaging. Processing and Analysis*; Academic Press series in biomedical engineering, pp. 393-420. Academic Press, San Diego, CA, U.S.A.
- [25] Thyssen, A. (2017). *ImageMagick v6 Examples - Resampling Filters*. <http://www.imagemagick.org/Usage/filter> (Accessed: January 25, 2017).
- [26] Turkowski, K. (1990). *Filters for Common Resampling Tasks*. In Glassner, A.S. (ed.), *Graphics Gems*, pp. 147-165. Academic Press, San Diego, CA, U.S.A.
- [27] Unser, M. (1999). *Splines: A Perfect Fit for Signal and Image Processing*. IEEE Signal Process. Mag., 16, pp. 22-38.
- [28] WOLFRAM (2017). *Filter-Design Window Functions*. <https://reference.wolfram.com/language/guide/WindowFunctions.html>. (Accessed: January 25, 2017).
- [29] Zaghetto, A., De Queiroz, R., L. (2007). *MRC Compression of Compound Documents Using H.264/AVC-I*. Simpósio Brasileiro de Telecomunicações, Recife, Brasil.
- [30] Zaghetto, A., De Queiroz, R.L. (2008). *Iterative Pre- and Post-Processing for MRC Layers of Scanned Documents*. In Proceedings of the 15th IEEE International Conference on Image Processing, pp. 1009-1012, San Diego, CA, U.S.A.